# Real-time implementation of frequency adaptive transforms

Nicki Holighaus
joint work with G.A. Velasco, M. Dörfler and T. Grill

Acoustics Research Institute, Austrian Academy of Sciences
Numerical Harmonic Analysis Group (NuHAG), Universität Wien
Austrian Research Insitute for Artificial Intelligence (OFAI)

8 December 2012

# Outline

# Outline

## Nonstationary Gabor transform (Fourier version)

The finite, discrete Nonstationary Gabor Transform (NSGT) of a signal $f$ of length $L$, with window functions $\{g_k\}_k$ is given by

$$c_{n,k} = \sum_{j=0}^{L-1} \hat{f}[j]\overline{g_k[j]}e^{\frac{2\pi i n a_k j}{L}} = \sum_{l=0}^{L-1} f[l]\overline{\check{g}_k[l - na_k]}. \tag{1}$$

- If $\mathcal{G}(\mathbf{g}, \mathbf{a}) := \{g_k[\cdot]\, e^{-\frac{2\pi i n a_k \cdot}{L}}\}_{n,k}$ spans $\mathbb{C}^L$, we say that they form a frame for $\mathbb{C}^L$.
- A Fourier-side NSGT is a (sampled), possibly non-uniform filterbank.

# NSG algorithms

**NSG analysis: $c = \mathbf{NSGTf}_L(f, \mathbf{g}, \mathbf{a})$**

1: **Initialize** $f, g_k$ for all $k \in I_K$
2: $f \leftarrow \mathbf{FFT}_L(f)$
3: **for** $k \in I_K,\ n = 0, \ldots, L/a_k - 1$ **do**
4: $\quad c_k \leftarrow \sqrt{L/a_k} \cdot \mathbf{IFFT}_{L/a_k}(f\overline{g_k})$
5: **end for**

**NSG synthesis: $\tilde{f} = \mathbf{iNSGTf}_L(c, \tilde{\mathbf{g}}, \mathbf{a})$**

1: **Initialize** $c_{n,k}, \tilde{g_k}$ for all $n = 0, \ldots, L/a_k - 1,\ k \in I_K$
2: **for** $k \in I_k$ **do**
3: $\quad f_k \leftarrow \sqrt{a_k/L} \cdot \mathbf{FFT}_{L/a_k}(c_k)$
4: **end for**
5: $\tilde{f} \leftarrow \sum_{k \in I_K} f_k \tilde{g_k}$
6: $\tilde{f} \leftarrow \mathbf{IFFT}_L(\tilde{f})$

# Outline

# Consideration I - FIR filters

Compact time support allows for segment-wise convolution, either direct or via FFT convolution (using overlap-add/overlap-save).

$\Rightarrow$ Exact coefficients can be obtained segmentwise.

Direct convolution is fast whenever a good downsampling strategy is involved, e.g. fast Wavelet transform.

# Consideration I - FIR filters II

Direct convolution: $LN/D$ multiplications per filter, where $D$ is the downsampling factor and $N$ the filter impulse response length in samples. Additional book-keeping, if many filters with different impulse response length are involved.
Fast convolution: At least $L$ multiplications per filter (more if performed segmentwise), since FIR filters cannot be bandlimited. Additionally, Fourier transforms of each segment are required.

$\Rightarrow$ FIR techniques are efficient for a moderate number of filters using well-structured downsampling rates. This limits flexibility, e.g. choice of redundancy.

Frame operator structure is not immediately apparent, invertibility cannot be determined easily.

$\Rightarrow$ In some cases (remember Monika's talk!), mathematicians say we can still determine invertibility relatively easily, by going to the Fourier side.

# Consideration II - Bandlimited filters

Fourier transform(s) (of each segment) are required, but afterwards filtering amounts to $M$ (filter bandwidth) multiplications per filter.

Downsampling rates can be chosen freely, i.e. any rational factor $D$ ($\geqslant 1$) such that $L/D$ is integer is admissible. Varying downsampling factors have little impact on computational complexity.

$\Rightarrow$ Efficient also for large numbers of filters, flexible redundancy.

# Consideration II - Bandlimited filters II

The Fourier-side frame operator is diagonal (or diagonal dominant) if the downsampling factors are small enough. Consequently, invertibility can be determined fairly easily. This is also referred to as *painless* and *almost painless* case.

$\Rightarrow$ Allows for fast direct/iterative reconstruction techniques.

A full-length FFT is required to obtain the coefficients, preventing real-time applications. Segmenting *always* introduces aliasing into the coefficients.

$\Rightarrow$ Any segmenting technique will corrupt the coefficients.

# Consideration III - What we desire? Everything!

- Efficient and perfect reconstruction
- Potential real-time (bounded delay) processing
- Fast algorithms for a large number of filters of various length and flexible downsampling factors
- Coefficients of a segmented transform should resemble the full length transform in both structure and information content
- Synthesis from modified coefficients should resemble the full length transform (this *is* different from the previous point)
- Ideally the technique would be applicable for a large variety of transform designs

# Outline

# Sliced NSG filterbanks

Frequency adaptive NSG filterbanks with bandlimited windows and perfect reconstruction are designed easily, e.g. constant-Q and ERBlet frames are available.

Step 1: Segment the signal into slices of (for now) uniform length $L_s$ with half-overlap

Step 2: Weigh the segments by a partition of unity, comprised of compactly supported functions

Step 3: Apply the same (for now) NSG filterbank to each weighted slice

# Sliced NSG filterbanks

Frequency adaptive NSG filterbanks with bandlimited windows and perfect reconstruction are designed easily, e.g. constant-Q and ERBlet frames are available.

Step 1: Segment the signal into slices of (for now) uniform length $L_s$ with half-overlap

Step 2: Weigh the segments by a partition of unity, comprised of compactly supported functions

Step 3: Apply the same (for now) NSG filterbank to each weighted slice

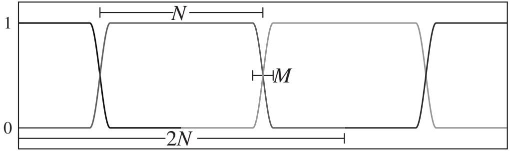



Figure: Slicing by a partition of unity, zero-padded to a half-overlap situation.

# Sliced NSG filterbanks

Frequency adaptive NSG filterbanks with bandlimited windows and perfect reconstruction are designed easily, e.g. constant-Q and ERBlet frames are available.

Step 1: Segment the signal into slices of (for now) uniform length $L_s$ with half-overlap

Step 2: Weigh the segments by a partition of unity, comprised of compactly supported functions

Step 3: Apply the same (for now) NSG filterbank to each weighted slice

Each (weighted) slice can be reconstructed perfectly, as the full signal before. The sum of correctly placed slices equals the full signal.
$\Rightarrow$ Perfect reconstruction is still possible.

Analysis and synthesis delay amount to 1 slice length, plus processing time.

What about the other properties?

# Desired properties I - NSG properties

As for

- efficient perfect reconstruction and
- fast algorithms for a large number of filters of various length and downsampling factors,

those properties are inherited directly from the corresponding NSG filterbank for signals of length $L_s$.

Redundancy, however, is approximately doubled, due to half overlap of the slices, although the effect on computation time is small.

# Desired properties II - Real-time and independence

As for

- potential real-time (bounded delay) processing and
- being suitable for a large variety of transform designs,

each slice can be transformed and synthesized individually, allowing for processing with bounded delay and in linear time. Any NSG filterbank that can be defined for length $L_s$ is admissible.

## Desired properties II - Real-time and independence

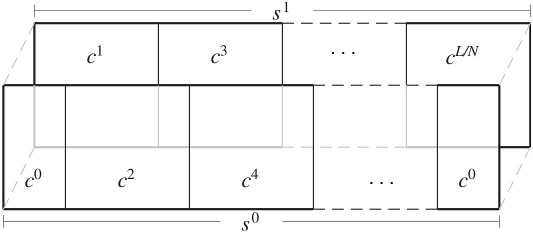
As for

- coefficients of a segmented transform should resemble the full length transform in both structure and information content and
- synthesis from modified coefficients should resemble the full length transform:

These properties are more involved and closely connected to both the time-localization of the analysis, resp. synthesis atoms of the frame, as well as the properties and “correct” interpretation of the *slicing*.

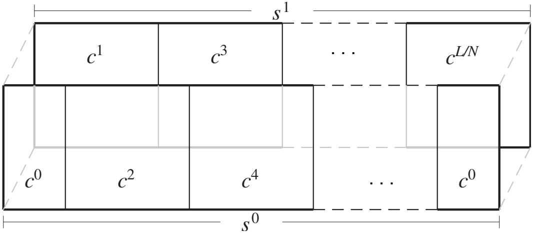
# Intuition I - Interpretation of the coefficients

Half overlap in the slicing step leads to a situation where 2 frame elements are located at the same time-frequency position in relation to the full signal. The time-frequency plane as "spanned" by the slices can be visualized as 2-layered (irregular) array, with the slices alternating between layers.




Since the slicing windows form a partition of unity, we expect the sum of the 2 layers to approximate the coefficients of some *corresponding* full length transform. This can be made explicit.

# Technicalities I - Preparation




- Rearrange the slice coefficients in 2 layers: For $l = \mathrm{mod}(m, 2)$, $k \in I_K$ and $n^s = 0, \ldots, L_s/a_k - 1$, let

$$s^l_{n^s+(m-1)L_s/(2a_k),k} \longleftarrow c^m_{n^s,k} \tag{2}$$

- Things are easier when slice length and number of time steps in each frequency band is even.
- In the following, let $N := L_s/2$. $I_K$ is the index set for the filters.

# Technicalities II

## Proposition

*Let $\mathcal{G}(\mathbf{g}^{\mathcal{L}}, \mathbf{a})$ be a nonstationary Gabor system for $\mathbb{C}^L$. Further, let $h_m = \mathbf{T}_{mN} h_0$ with $\sum_{m=0}^{L/N-1} h_m \equiv 1$ and define $g_k \in \mathbb{C}^{2N}$, for all $k \in I_K$ by*

$$g_k[j] = g_k^{\mathcal{L}}[jL/(2N)].$$

*For $f \in \mathbb{C}^L$, denote by $c \in \mathbb{C}^{L/a_k \times |I_K|}$ the CQ-NSGT coefficients of $f$ with respect to $\mathcal{G}(\mathbf{g}^{\mathcal{L}}, \mathbf{a})$ and by $s \in \mathbb{C}^{2 \times L/a_k \times |I_K|}$ the sliCQ coefficients of $f$ with respect to $h_0$ and $\mathcal{G}(\mathbf{g}, \mathbf{a})$. Then*

$$\begin{aligned} &\left| s^0_{n,k} + s^1_{n,k} - c_{n,k} \right| \\ &\leqslant \|f\|_2 \Big( \|(1 - h_0 - h_1) \mathbf{T}_{n^s a_k} \breve{g}_k^{\mathcal{L}} \|_2 \\ &+ \|(h_0 + h_1) \sum_{j=1}^{\frac{L}{2N}-1} \mathbf{T}_{n^s a_k + 2jN} \breve{g}_k^{\mathcal{L}} \|_2 \Big) \end{aligned} \tag{3}$$

*for $n = mN/a_k + n^s$, with $m = 0, \ldots, L/N - 1$ and $n^s = 0, \ldots, N/a_k - 1$.*

# Technicalities III - The error terms

Actually, the the error estimate is quite coarse, yet it gives the right idea:

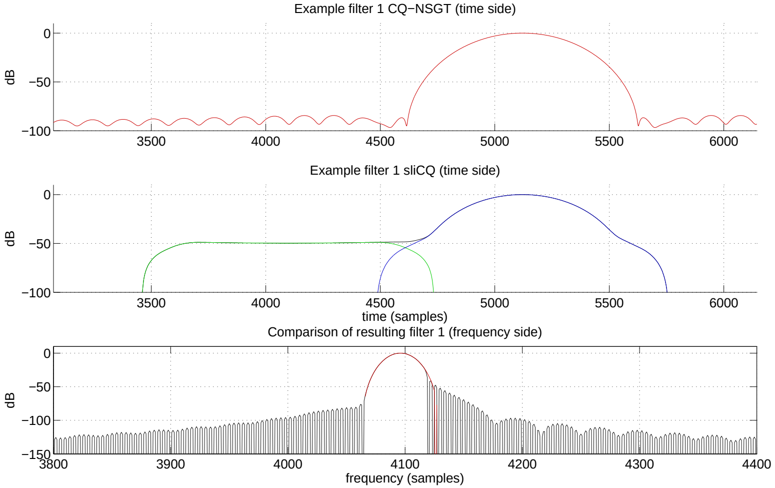
- $\|(1 - h_0 - h_1)\mathbf{T}_{n^s a_k}\breve{g}_k^{\mathcal{L}}\|_2$

  This term contains the error originating from cutting $\breve{g}_k^{\mathcal{L}}$ with the slicing windows $h_m + h_{m+1}$. It will be small if $\mathbf{T}_{n^s a_k}\breve{g}_k^{\mathcal{L}}$ is well concentrated in the interval, where $h_m + h_{m+1} = 1$.

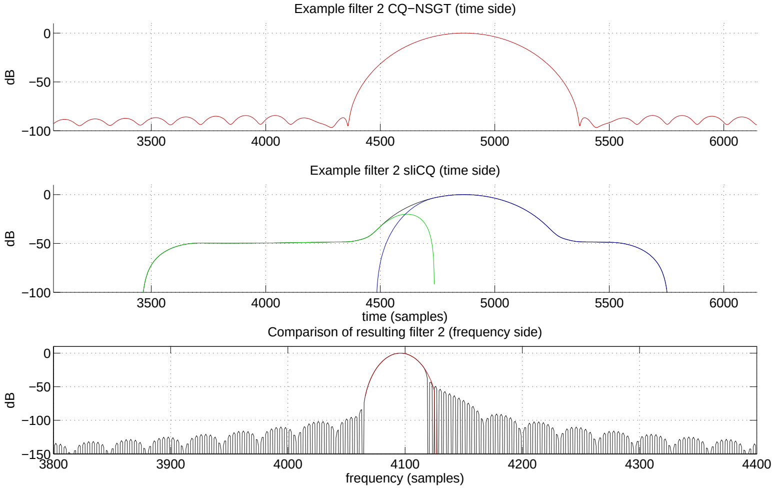
- $\|(h_0 + h_1)\sum_{j=1}^{\frac{L}{2N}-1}\mathbf{T}_{n^s a_k + 2jN}\breve{g}_k^{\mathcal{L}}\|_2$

  This term contains the aliasing induced by subsampling the Fourier transform $g_k^{\mathcal{L}}$ of $\breve{g}_k^{\mathcal{L}}$ to be of length $2N$. It will be small if $\breve{g}_k^{\mathcal{L}}$ is well concentrated and the slices are zero-extended enough.

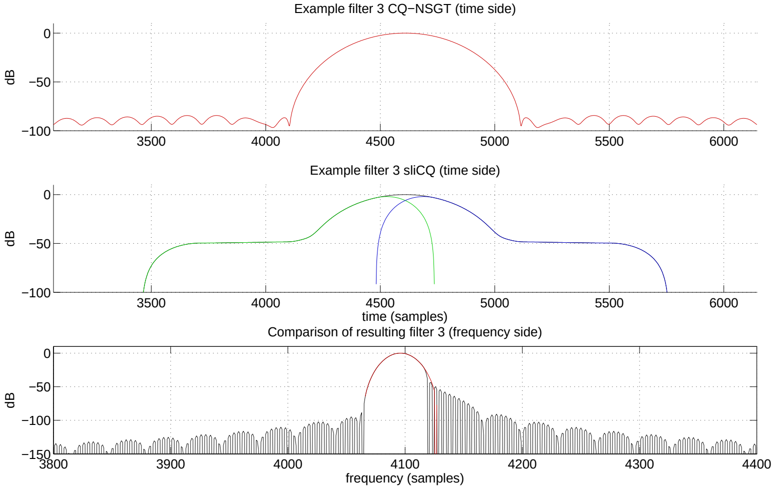
# The resulting analysis filters - worst case



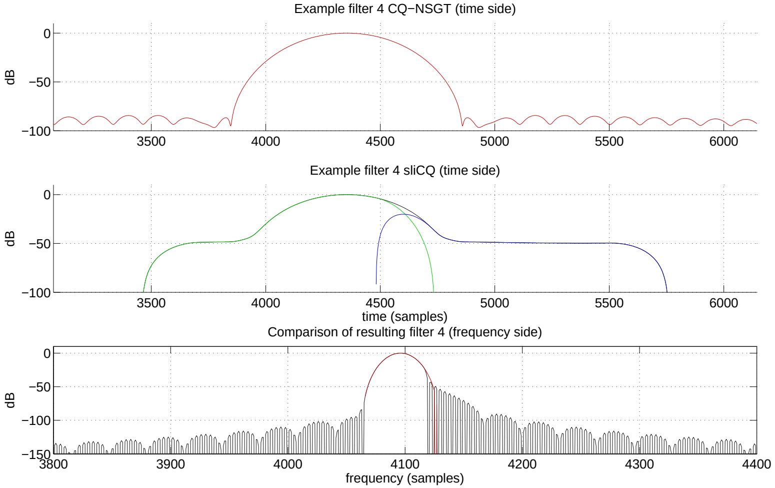
# The resulting analysis filters - worst case



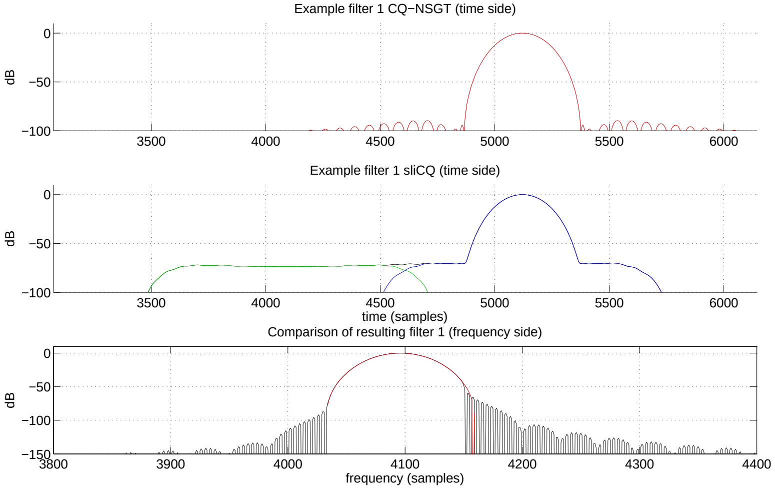
# The resulting analysis filters - worst case



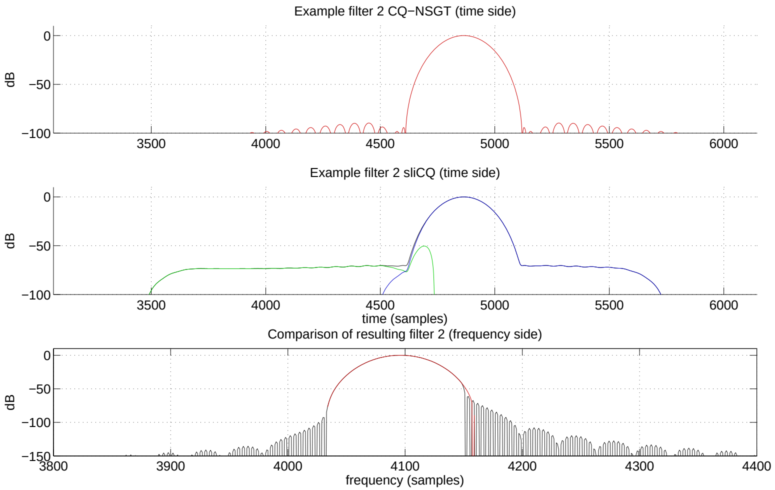
# The resulting analysis filters - worst case



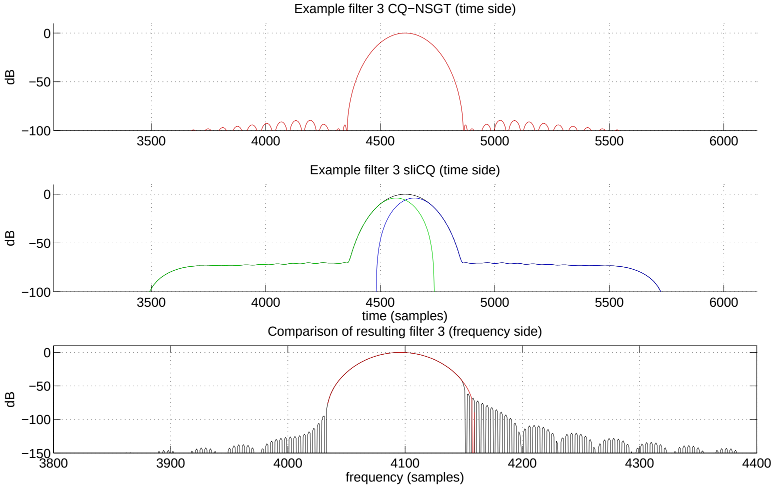
# The resulting analysis filters - double bandwidth



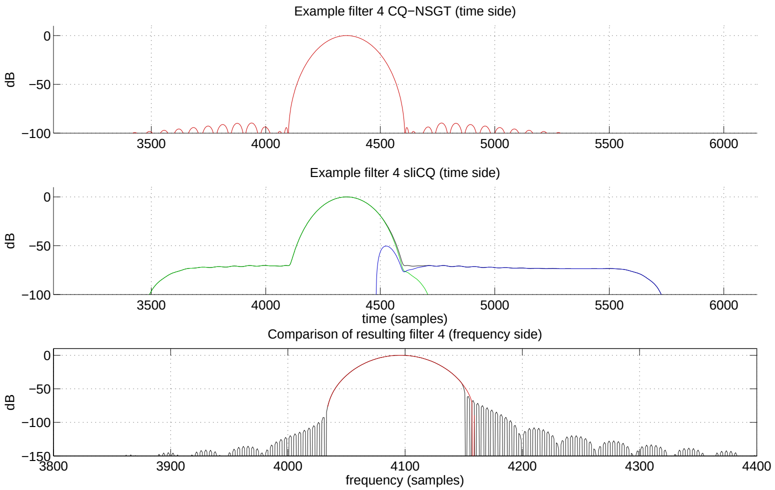
# The resulting analysis filters - double bandwidth



# The resulting analysis filters - double bandwidth



# The resulting analysis filters - double bandwidth



# Intuition II - Synthesis I

In the case of unmodified coefficients, the perfect reconstruction property is unaffected.

Intuition: Slicewise synthesis from modified coefficients produces results close to the full length transform whenever the elements of the dual frame are largely unaffected by truncating them at the slice ends.

Consequently, the dual frame should consist of elements well-localized in time, ideally the frame would be close to or tight.

# Intuition II - Synthesis I

In the case of unmodified coefficients, the perfect reconstruction property is unaffected.

Intuition: Slicewise synthesis from modified coefficients produces results close to the full length transform whenever the elements of the dual frame are largely unaffected by truncating them at the slice ends.

Consequently, the dual frame should consist of elements well-localized in time, ideally the frame would be close to or tight.

To be honest, the synthesis step could still be improved, with “tighter” frames. The experimental results are quite acceptable, though.

# Intuition II - Synthesis II

To prevent blocking artifacts, we prefer using smooth *dual slicing windows*, such that

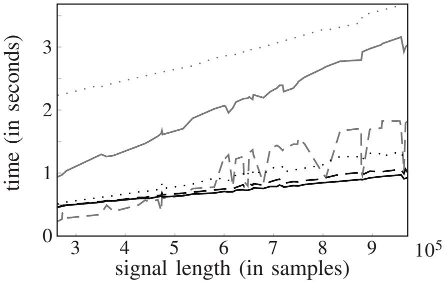
$$\sum_{m=0}^{L/N-1} \mathbf{T}_{mN} \left( h_0 \overline{\tilde{h}_0} \right) \equiv 1. \tag{4}$$

This additional step improves perceptual synthesis quality, especially when the dual frame elements are not sufficiently well localized.

As long as the equation above is satisfied, perfect reconstruction is retained.

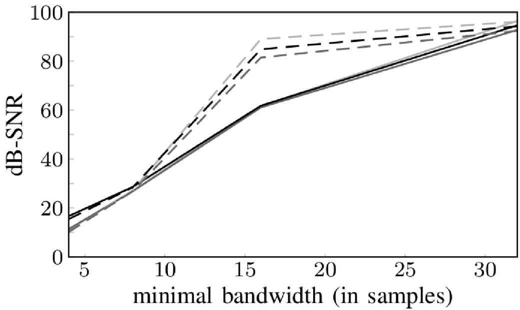
# Outline

## Computation time



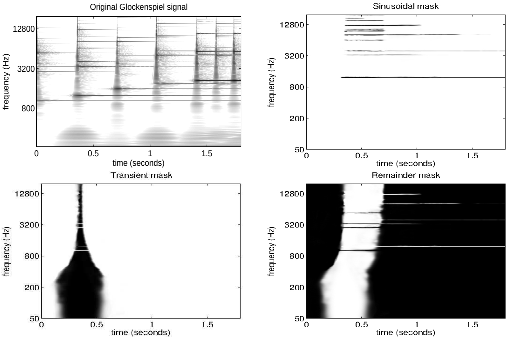

We show computation time versus signal length of the CQ transform (dotted gray), CQ-NSGT (dashed gray) and various sliCQ transforms. The sliCQ transforms were taken with slice lengths 4096 (solid gray), 16384 (dotted black), 32768 (dashed black) and 65536 (solid black) samples.

## Coefficient approximation



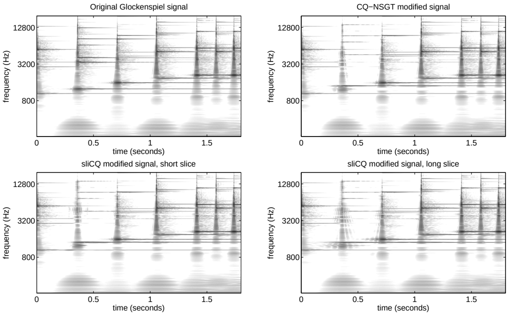

We show the SliCQ coefficient approximation error against the minimal admissible bandwidth for a set of random signals. All transforms use Blackman-Harris windows. Solid and dashed lines represent long (1/4 slice length) and short (1/128 slice length) transition areas respectively, while colors correspond to the slice length: 4096 (light gray), 16384 (dark gray) and 65536 samples (black).

# Processing experiment - The masks




The signal is masked with each of the displayed masks separately. The coefficients multiplied with the sinusoid mask are shifted upwards by 8 bins, corresponding to a transposition by 2 semitones.

# Processing experiment - The transposed signal




The resulting coefficients from all 3 maskings are summed, then synthesized. The resulting, re-analyzed signals are shown above. Note that the masks chosen were identical in all 3 displayed cases.

## Possible extensions

- variant-NSGTs for various frequency scales (Mel, Bark, etc) can be realized easily
- true *bounded delay* implementation

# Possible extensions

- variant-NSGTs for various frequency scales (Mel, Bark, etc) can be realized easily
- true *bounded delay* implementation
- adaptivity of slice lengths

# Possible extensions

- variant-NSGTs for various frequency scales (Mel, Bark, etc) can be realized easily
- true *bounded delay* implementation
- adaptivity of slice lengths
- higher subsampling factors in the NSG step, see Thibaud's talk (current implementation is *painless*)

# Possible extensions

- variant-NSGTs for various frequency scales (Mel, Bark, etc) can be realized easily
- true *bounded delay* implementation
- adaptivity of slice lengths
- higher subsampling factors in the NSG step, see Thibaud's talk (current implementation is *painless*)
- improve the frame bounds with better, e.g. warped, windows

# Possible extensions

- variant-NSGTs for various frequency scales (Mel, Bark, etc) can be realized easily
- true *bounded delay* implementation
- adaptivity of slice lengths
- higher subsampling factors in the NSG step, see Thibaud's talk (current implementation is *painless*)
- improve the frame bounds with better, e.g. warped, windows
- and probably more...

# Possible extensions

- variant-NSGTs for various frequency scales (Mel, Bark, etc) can be realized easily
- true *bounded delay* implementation
- adaptivity of slice lengths
- higher subsampling factors in the NSG step, see Thibaud's talk (current implementation is *painless*)
- improve the frame bounds with better, e.g. warped, windows
- and probably more...

Thank you for your attention!